

# Présentation des Données du Web - TP 3 : XML Relationnel

TD en binômes à rendre le 25 octobre.

## 1) Oracle-XML : CLOB vs Binary XML

1. Récupérez les documents XML que vous avez défini pour l'exercice 1 du TD1 (bâtiment et presse).
2. Connectez vous à ORACLE avec le commande `rlwrap sqlplus login/mot_de_passe@venus/master`
3. Activez la visualisation des temps d'exécution des requêtes avec `SET TIMING ON`.
4. Créez des tables pour enregistrer vos documents XML  

```
CREATE TABLE nom_de_la_table_CLOB (nom_colonne_texte varchar(20), nom_colonne_xml XMLTYPE)  
XMLTYPE nom_colonne_xml STORE AS CLOB;  
CREATE TABLE nom_de_la_table_binaryxml (nom_colonne_texte varchar(20), nom_colonne_xml XMLTYPE)  
XMLTYPE nom_colonne_xml STORE AS BINARY XML;
```
5. Enregistrez vos documents XML dans la base avec l'instruction  

```
INSERT INTO nom_de_la_table (nom_colonne_texte,nom_colonne_xml)  
VALUES ('test', sys.xmltype.createxml('<mettreXMLici/>')) );
```
6. Pour chaque document XML, exécutez trois requêtes XPath (de complexité différente) à votre choix. Utilisez les instructions de base suivantes.  

```
SELECT EXTRACT(nom_colonne_xml,'/mettre/ici/la/requête/XPath') FROM nom_de_la_table;  
SELECT XMLQUERY('/mettre/ici/la/requête/XQuery' PASSING nom_colonne_xml RETURNING CONTENT)  
FROM nom_de_la_table;
```

## 2) Stockage schema-unaware : EDGE vs Vertical-EDGE vs Monet (sur les batiments)

1. Créez des tables en SQL pour implementer le stockage EDGE, Vertical-EDGE et Monet (voir le cours).
2. Remplissez les tables avec des valeurs correspondants à votre document XML pour les batiments.
3. Traduisez en SQL les requêtes que vous avez choisi dans l'exercice précédent, pour chacun des schémas de stockage. Si cela n'est pas possible, simplifiez vos requêtes.

## 3) Stockage schema-aware (sur la presse)

1. Considérez la DTD du TD1 pour la presse, et donnez le schéma de stockage pour cette DTD.
2. Remplissez les tables avec des valeurs correspondant à votre document XML pour la presse.
3. Traduisez en SQL les requêtes choisies dans l'exercice précédent. Simplifiez vos requêtes, si nécessaire.
4. Proposez une requête XMLAGG permettant de réexporter les données de la table relationnelle en XML.

## 4) Interval-encoding avec SAX

1. Donnez l'encodage begin/end de l'arbre 'batiments', puis enregistrez-le dans la table NODE.
2. À l'aide de la classe SaxParser (.java sur Moodle), programmez l'encodage par intervalles begin/end. Cette classe permet de parser un document XML en streaming (noeud après noeud). Modifiez les méthodes `startElement` et `endElement` qui sont déclenchées à l'ouverture et fermeture d'un noeud de type `element`, ainsi que la méthode `characters` qui est déclenchée lorsque on rencontre un noeud de texte. L'output du programme sera une liste de mises à jour pour la table NODE, comme la suivante.  

```
INSERT INTO NODE (id,begin,end,parent,tag,nodtyp) VALUES(0,1,16,null,'racine','element')
```
3. Testez le programme sur le document XML concernant la presse, et traduisez les requêtes que vous avez proposé dans l'exercice 1 en SQL pour la table NODE.
4. Testez le programme avec le fichier XML <http://www.ins.cwi.nl/projects/xmark/Assets/standard.gz> et reportez le temps d'exécution.

(extra) Modifier le programme afin d'implémenter l'encodage pre/post du document.